Training a Large Language Model from Scratch: A Research Based Assessment

Yash Kambli
yash.kambli22@spit.ac.in

Ayush Nemade
ayush.namade22@spit.ac.in

Abstract

This assignment explores the process of building and training a Large Language Model (LLM) from scratch using a curated dataset of classic English literature. We analyze the impact of various hyperparameters and architectural decisions, including epoch count, learning rate, number of transformer layers, attention heads, and critical transformer components. The experiments are designed to observe how these factors influence training dynamics, validation loss, and model output quality. Our findings offer empirical insights into the interplay between LLM architecture and training stability, contributing to a practical understanding of LLM development and optimization.

1 Introduction

Large Language Models (LLMs) have significantly advanced the field of natural language processing by enabling machines to understand and generate text that closely resembles human writing. While most LLMs today are pre-trained on massive corpora, understanding the intricacies of training such models from scratch provides valuable insight into model behavior and design decisions. This assignment focuses on training an LLM from scratch and analyzing how architectural choices and hyperparameter variations impact training effectiveness, model generalization, and output quality.

2 Dataset and Preprocessing

We use a dataset comprising three classic novels sourced from Project Gutenberg:

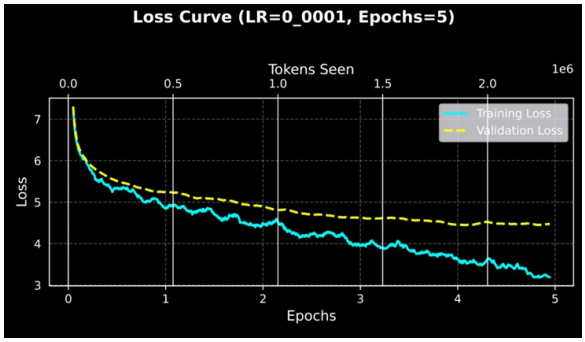
- *Pride and Prejudice* by Jane Austen
- *Dracula* by Bram Stoker
- *A Room with a View* by E.M. Forster

The texts are concatenated into a single corpus, tokenized into subword units using a custom vocabulary, and segmented into input-target sequences suitable for next-token prediction. We divide the corpus into a 90:10 train-validation split. This dataset provides rich syntactic diversity and literary style ideal for testing LLM capabilities on structured, formal language.

3 Experiments and Results

3.1 Baseline Training

Configuration: 5 epochs, learning rate = 0.0001, 12 transformer layers, 12 attention heads



(a) Training and Validation Loss

```
The tree was a
in the morning, and the two ladies were not the
consequence. The Professor had been a little, and
```

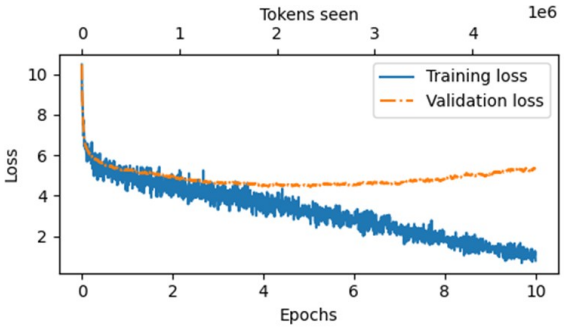
(b) Generated Output Sample

Figure 1: Baseline Configuration

Results: Training Loss = 3.0843, Validation Loss = 4.4537, Training Time = 16.53 mins
Inference: The baseline model demonstrates reasonable convergence with stable validation performance. This setup is used as a reference for all subsequent experiments.

3.2 Experiment 1: Varying Epochs and Learning Rate

Epochs = 10, Learning Rate = 0.0001: Training loss improved slightly, but validation loss stagnated after epoch 2, indicating early overfitting.



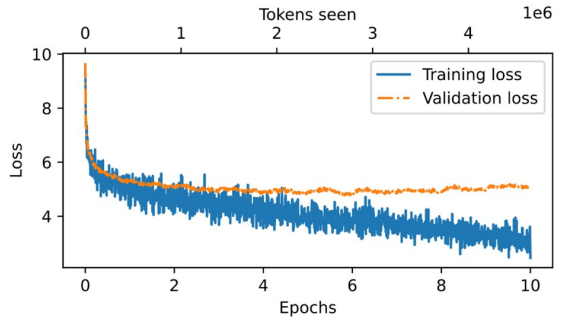
(a) Training and Validation Loss

```
Output text:
 The tree was a smile as Miss Bartlett.

"Oh," said Miss Bartlett. "I am afraid
```

(b) Generated Output Sample

Figure 2: Epochs: 10, Learning Rate: 0.0001

Epochs = 10, Learning Rate = 0.0005: Faster training convergence, but less improvement in validation loss. Model started to overfit around epoch 8.



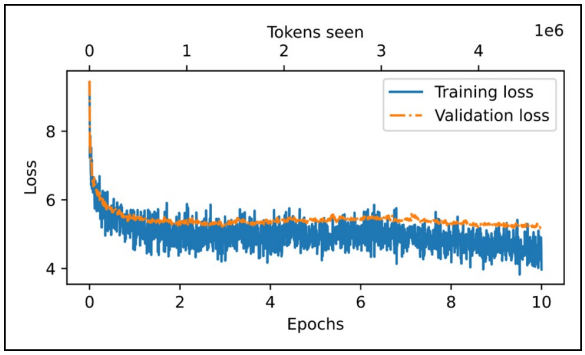
(a) Training and Validation Loss

```
The tree was a
hills of terror and the room, and the
hills of the chin, and the chin, and the
```

(b) Generated Output Sample

Figure 3: Epochs: 10, Learning Rate: 0.0005

Epochs = 10, Learning Rate = 0.001: Training loss reduced quickly, but validation loss worsened significantly. Output was incoherent.



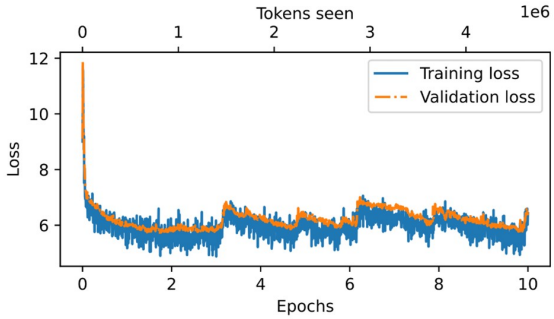
(a) Training and Validation Loss

```
The tree was
the

the

"I have
```

(b) Generated Output Sample

Figure 4: Epochs: 10, Learning Rate: 0.001

Epochs = 10, Learning Rate = 0.01: The model diverged early; loss spiked and output degraded completely.



(a) Training and Validation Loss

The tree was

(b) Generated Output Sample

Figure 5: Epochs: 10, Learning Rate: 0.01

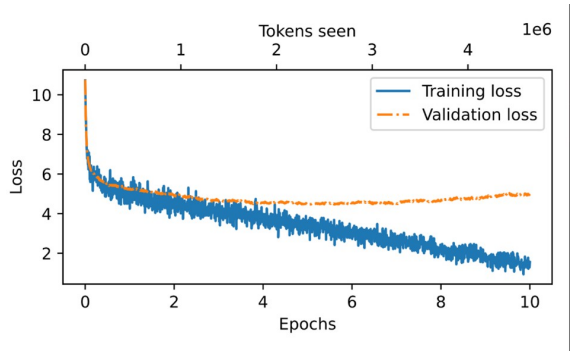
Inference: A learning rate of 0.0001 provided the best balance. Larger values led to instability and poor generalization.

Table 1: Effect of Learning Rate with Epochs = 10

Learning Rate	Training Loss	Validation Loss	Notes
0.0001	2.5432	4.5001	Best generalization
0.0005	1.9876	5.1032	Overfits after 8 epochs
0.001	1.2231	6.6720	Unstable
0.01	Diverged	Diverged	Training collapsed

3.3 Experiment 2: Varying Transformer Layers

7 Layers: Training Loss = 1.5563, Validation Loss = 4.9890. Output was moderately coherent.



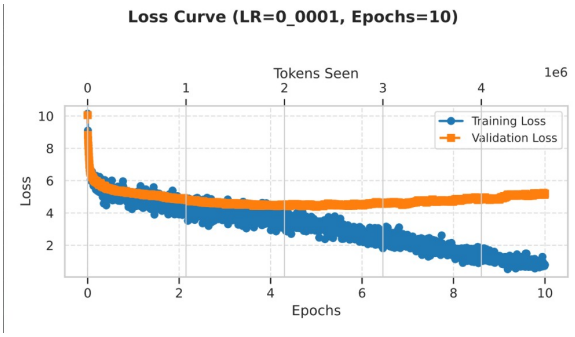
(a) Training and Validation Loss

```
The tree was a very different manner, and the very
drew into the very large and the way. The man, and was all
```

(b) Generated Output Sample

Figure 6: Number of Layers: 7

17 Layers: Training Loss = 0.7859, Validation Loss = 5.1992. Deeper model learned more but started to overfit.



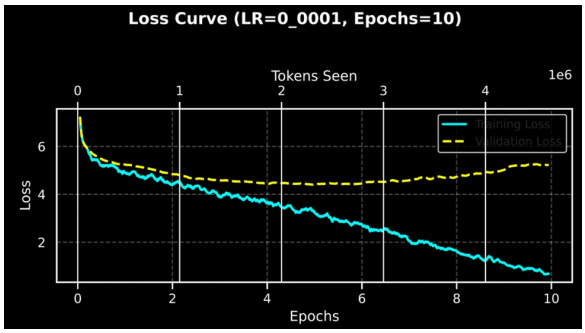
(a) Training and Validation Loss

```
The tree was the same moment, that we got into the room.

I had a moment I could not make sure that he would
```

(b) Generated Output Sample

Figure 7: Number of Layers: 17

20 Layers: Training Loss = 0.6342, Validation Loss = 5.1871. Text was more context-aware; overfitting evident.



(a) Training and Validation Loss

```
The tree was the same parental care.

"I wish I could get on this point," said Van
Hels
```

(b) Generated Output Sample

Figure 8: Number of Layers: 20

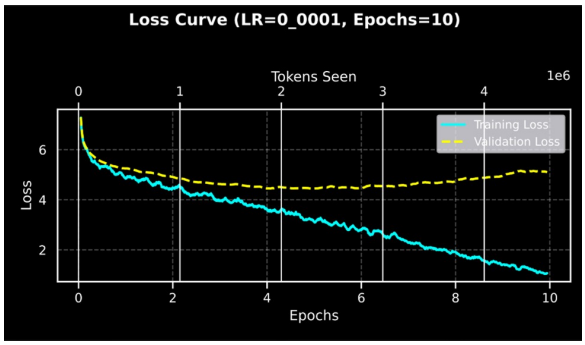
Inference: Increasing transformer depth improved training loss but slightly worsened validation, suggesting diminishing returns and risk of overfitting.

Table 2: Effect of Transformer Depth (LR = 0.0001, Epoch = 10)

Layers	Training Loss	Validation Loss	Output Quality
7	1.5563	4.9890	Acceptable
17	0.7859	5.1992	Improved
20	0.6342	5.1871	Best

3.4 Experiment 3: Varying Attention Heads

6 Heads: Training Loss = 0.8340, Validation Loss = 5.1018. Slightly less expressive.



(a) Training and Validation Loss

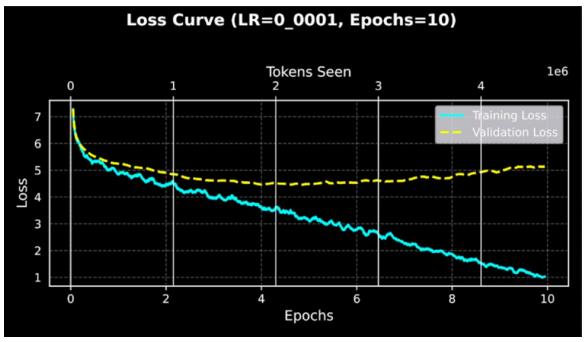
```
The tree was the first; but he had for his mind,
his palms together and was not so much rather than that, his manners
```

(b) Generated Output Sample

Figure 9: Number of Attention Heads: 6

12 Heads (Baseline): Balanced training and validation loss. Refer Figure 1 for Training and Validation Loss plot and Generated Output Sample.

16 Heads: Training Loss = 0.7713, Validation Loss = 5.1355. Slightly better than 6, but not significantly.



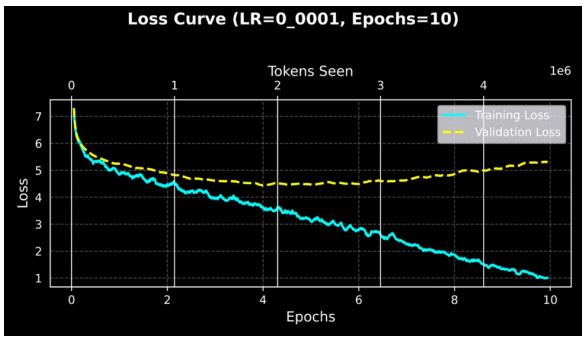
(a) Training and Validation Loss

```
The tree was a very pleasant."

"Well," said Mr. Bingley. "I am all
```

(b) Generated Output Sample

Figure 10: Number of Attention Heads: 16

24 Heads: Training Loss = 0.7319, Validation Loss = 5.2323. More heads didn't improve validation performance.



(a) Training and Validation Loss

```
The tree was not
proceeding to be too much for any disrespect to his aunt.

Miss Lavish; and, was
```

(b) Generated Output Sample

Figure 11: Number of Attention Heads: 24

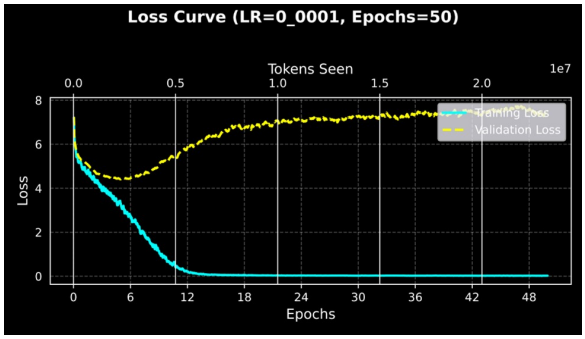
Inference: More attention heads improved training slightly but didn't help generalization.

Table 3: Effect of Attention Heads (Layers = 12, Epoch = 10, LR = 0.0001)

Attention Heads	Training Loss	Validation Loss	Comments
6	0.8340	5.1018	Less expressive
12	3.0843	4.4537	Balanced
16	0.7713	5.1355	Slight gain
24	0.7319	5.2323	No further gain

3.5 Extended Epochs with Best Config

Epochs = 50: Training Loss = 0.0232, Validation Loss = 7.4341



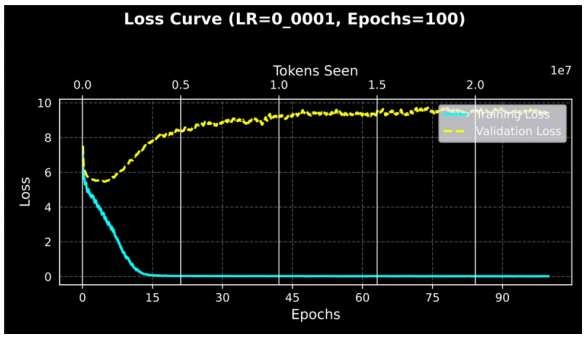
(a) Training and Validation Loss

```
The tree was laid down the room,
and that so we remained till at the moment.

I looked to raise that, I
```

(b) Generated Output Sample

Figure 12: Epochs = 50, Learning Rate = 0.0001

Epochs = 100: Training Loss = 0.0030, Validation Loss = 9.1142



(a) Training and Validation Loss

```
The tree was really
con. Mr. Hurst looked at her with the
day before, unable to the novelty of her manner.
```

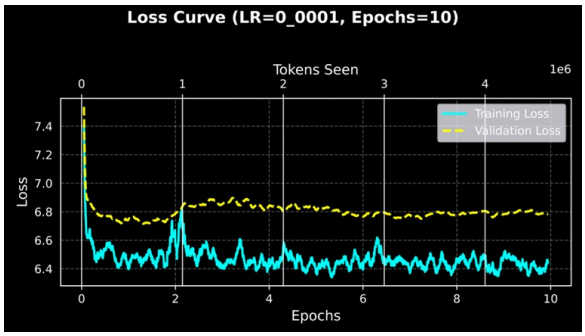
(b) Generated Output Sample

Figure 13: Epochs = 100, Learning Rate = 0.0001

Inference: Though training loss decreased, validation loss worsened significantly due to overfitting. Early stopping would be beneficial.

3.6 Experiment 4: Ablation Studies

No Normalization: Training Loss = 6.3594, Validation Loss = 6.7815. Model failed to converge meaningfully.



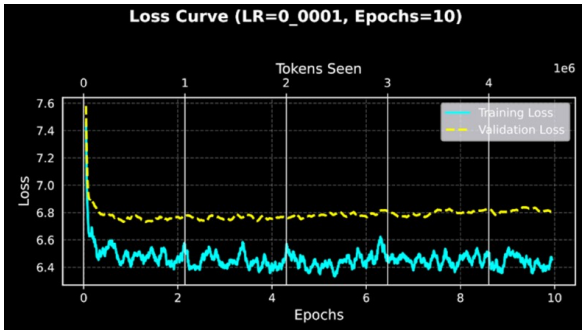
(a) Training and Validation Loss

```
The tree was
```

(b) Generated Output Sample

Figure 14: Removing normalization layers in all transformer layers

No Residual Connections: Training Loss = 6.3714, Validation Loss = 6.8086. Similar instability observed.



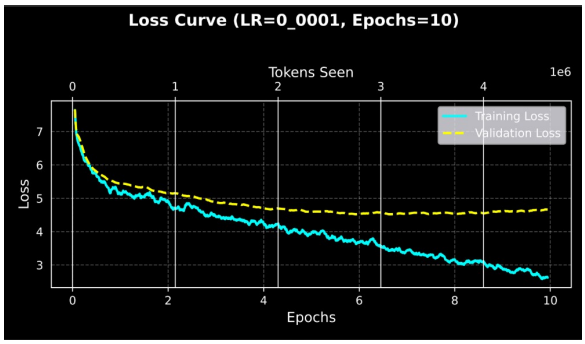
(a) Training and Validation Loss

The tree was

(b) Generated Output Sample

Figure 15: Removing shortcut (residual) connections in all transformer layers

No Feedforward Network: Training Loss = 2.6656, Validation Loss = 4.6727. Model partially trained but lacked abstraction capacity.



(a) Training and Validation Loss

The tree was a very kind of dance, and, the nose of the other.

"Look!"

(b) Generated Output Sample

Figure 16: Removing feedforward neural network in all transformer layers

Inference: Normalization and residuals are essential for stable training. Feedforward layers are also important but relatively less critical.

Table 4: Ablation Study Results (Epoch = 10, LR = 0.0001, Layers = 12, Heads = 12)

Removed Component	Training Loss	Validation Loss	Comments
Normalization	6.3594	6.7815	Highly unstable
Residuals	6.3714	6.8086	Unstable
Feedforward	2.6656	4.6727	Weakened capacity

4 Conclusion

Training an LLM from scratch offers significant insights into model dynamics and optimization strategies. Our experiments highlight the delicate balance between model complexity, training duration, and learning rate. While deeper models and more attention heads improve training performance, they increase the risk of overfitting. Ablation studies reaffirm the critical role of transformer components like normalization and residual connections in stabilizing training. Regularization and early stopping techniques are vital for training larger models over extended epochs.

5 Key Findings Summary

- **Learning Rate:** 0.0001 provided optimal balance between training speed and stability; higher rates (0.001, 0.01) led to training instability and poor generalization
- **Model Depth:** 20 transformer layers showed the best output quality with training loss of 0.6342, though with diminishing returns; deeper models improved training loss but risked overfitting
- **Attention Heads:** 12–16 heads offered the best performance-complexity tradeoff; increasing beyond 16 heads provided minimal benefits for validation performance
- **Training Duration:** Significant overfitting occurred beyond 50 epochs (validation loss: 7.4341); at 100 epochs, validation loss deteriorated to 9.1142 despite near-zero training loss
- **Critical Components:** Ablation studies confirmed that layer normalization and residual connections are essential for stable training, while feedforward networks contribute to model expressivity
- **Architecture Balance:** The ideal configuration balances complexity (depth and width) with regularization techniques to prevent overfitting on limited datasets